Amendments to the claims

1 (Original): A computer system having an interrupt handling apparatus, the computer system comprising:

one or more processors;

one or more hardware devices that are capable of interrupting the one or more processors using an interrupt signal;

an interrupt controller that is capable of handling the interrupts from the one or more hardware devices and capable of independently generating a low priority interrupt signal and a high priority interrupt signal for each processor;

the interrupt controller further comprising an enable device that is capable of enabling the interrupt signal independently for each processor and a priority device that is capable of assigning a priority to each interrupt signal destined for any processor wherein a particular interrupt signal is capable of being routed to either processor and is capable of being assigned a low priority or a high priority.

2 (Original): The system of claim 1, wherein the enable device further comprises a first processor enable device that is capable of enabling the interrupt signals for the first processor and a second processor enable device that is capable of enabling the interrupt signals for the second processor.

3 (Original): The system of claim 2, wherein each enable device comprises one or more flip-flips connected together with one or more logical gates.

4 (Original): The system of claim 3, wherein each enable device further comprises a register containing an interrupt enable signal that is fed into the logical gates in order to enable the interrupt signals of the particular processor.

5 (Original): The system of claim 1, wherein the priority device further comprises a first processor priority device that is capable of assigning a priority to the interrupt signals for the first processor and a second processor priority device that is capable of assigning a priority to the interrupt signals for the second processor.

Amendments to the claims

6 (Original): The system of claim 5, wherein each priority device comprises one or more flip-flips connected together with one or more logical gates.

7 (Original): The system of claim 6, wherein each priority device further comprises a register containing an interrupt priority signal that is fed into the logical gates in order to determine the priority for the interrupt signals of the particular processor.

8 (Original): The system of claim 7, wherein each priority device selects between a low priority signal and a high priority signal.

9 (Original): The system of claim 8, wherein the high priority signal comprises an FIQ signal and the low priority signal comprises an IRQ signal.

10 (Original): The system of claim 1, wherein the interrupt controller further comprises a device for multiplexing the one or more hardware interrupt signals with one or more software interrupt signals so that the processors are capable of being interrupted by hardware devices and software.

11 (Original): The system of claim 10, wherein the software interrupt signals comprise a software interrupt generated by one processor to interrupt another processor.

12 (Original): A computer system having an interrupt handling apparatus, the computer system comprising:

one or more processors;

one or more hardware devices that are capable of interrupting the one or more processors using an interrupt signal;

an interrupt controller that is capable of handling the interrupts from the one or more hardware devices and capable of independently generating a low priority interrupt signal and a high priority interrupt signal for each processor;

the interrupt controller further comprising means for independently enabling the interrupt

Amendments to the claims

signal for each processor and means for independently assigning a priority to each interrupt signal destined for any processor wherein a particular interrupt signal is capable of being routed to either processor and is capable of being assigned a low priority or a high priority.

13 (Original): The system of claim 12, wherein the enable means further comprises a first processor enable device that is capable of enabling the interrupt signals for the first processor and a second processor enable device that is capable of enabling the interrupt signals for the second processor.

14 (Original): The system of claim 13, wherein each enable device comprises one or more flip-flips connected together with one or more logical gates.

15 (Original): The system of claim 14, wherein each enable device further comprises a register containing an interrupt enable signal that is fed into the logical gates in order to enable the interrupt signals of the particular processor.

16 (Original): The system of claim 12, wherein the priority means further comprises a first processor priority device that is capable of assigning a priority to the interrupt signals for the first processor and a second processor priority device that is capable of assigning a priority to the interrupt signals for the second processor.

17 (Original): The system of claim 16, wherein each priority device comprises one or more flip-flips connected together with one or more logical gates.

18 (Original): The system of claim 17, wherein each priority device further comprises a register containing an interrupt priority signal that is fed into the logical gates in order to determine the priority for the interrupt signals of the particular processor.

19 (Original): The system of claim 18, wherein each priority device selects between a low priority signal and a high priority signal.

Amendments to the claims

20 (Original): The system of claim 19, wherein the high priority signal comprises an FIQ signal and the low priority signal comprises an IRQ signal.

21 (Original): The system of claim 12, wherein the interrupt controller further comprises a device for multiplexing the one or more hardware interrupt signals with one or more software interrupt signals so that the processors are capable of being interrupted by hardware devices and software.

22 (Original): The system of claim 21, wherein the software interrupt signals comprise a software interrupt generated by one processor to interrupt another processor.

23 (Currently amended): A method for interrupt handling in a multiple processor system, the method comprising:

assigning the interrupt signal to one of the multiple processors based on availability of each processor, wherein assigning the processor further comprises reading a register to determine the assignment of the interrupt signal to a particular processor;

assigning a priority level to the interrupt signal wherein the priority is selected from one or more different priorities; and

routing the interrupt signal with the assigned priority level to the assigned processor.

24 (Canceled).

25 (Currently amended): The method of claim ~~24~~ 23, wherein the register reading further comprises reading an enable register associated with each processor to determine the availability of each processor to handle the interrupt signal.

26 (Original): The method of claim 23, wherein the routing further comprises splitting up the servicing of the interrupt signal between the processors.

Amendments to the claims

27 (Original): The method of claim 23, wherein the processor assigning further comprises dynamically assigning the processor to handle a particular interrupt signal.

28 (Original): The method of claim 23, wherein assigning the priority further comprises reading a priority register to determine the priority of the interrupt signal.

29 (Original): The method of claim 28, wherein the priority assigning further comprises assigning a low priority to the interrupt signal and assigning a high priority to the interrupt signal.

30 (Original): The method of claim 23, wherein the priority assigning further comprises dynamically assigning the priority to the interrupt signal.

31 (Currently amended): A method for interrupt handling in a multiple processor system, the method comprising:

assigning the interrupt signal to one of the multiple processors based on availability of each processor, wherein assigning the processor further comprises reading a register to determine the assignment of the interrupt signal to a particular processor; and

assigning a priority level to the interrupt signal wherein the priority is selected from one or more different priorities.

32 (Canceled).

33 (Currently amended): The method of claim ~~32~~ 31, wherein the register reading further comprises reading an enable register associated with each processor to determine the availability of each processor to handle the interrupt signal.

34 (Original): The method of claim 31, wherein the routing further comprises splitting up the servicing of the interrupt signal between the processors.

Amendments to the claims

35 (Original): The method of claim 31, wherein the processor assigning further comprises dynamically assigning the processor to handle a particular interrupt signal.

36 (Original): The method of claim 31, wherein assigning the priority further comprises reading a priority register to determine the priority of the interrupt signal.

37 (Original): The method of claim 36, wherein the priority assigning further comprises assigning a low priority to the interrupt signal and assigning a high priority to the interrupt signal.

38 (Original): The method of claim 31, wherein the priority assigning further comprises dynamically assigning the priority to the interrupt signal.

39 (Original): A computer system, comprising:

a first processor;

a second processor;

a peripheral controller having a first bus that is capable of connecting the first processor to a set of peripheral resources and a second bus that is capable of connecting the second processor to the same set of peripheral resources wherein each peripheral in the set of peripheral resources is capable of generating an interrupt signal;

the peripheral controller further comprising an interrupt controller that is capable of handling the interrupts from the set of peripheral resources and software interrupts and capable of independently generating a low priority interrupt signal and a high priority interrupt signal for each processor from the peripheral and software interrupts, the interrupt controller further comprising a first processor enable device that is capable of enabling the interrupt signals for the first processor, a second processor enable device that is capable of enabling the interrupt signals for the second processor, a first processor priority device that is capable of assigning a priority to the interrupt signals for the first processor and a second processor priority device that is capable of assigning a priority to the interrupt signals for the second processor wherein a particular interrupt signal is capable of being routed to either processor and is capable of being assigned a

Amendments to the claims

low priority or a high priority.

40 (Original): A computer system, comprising:

a first processor;

a second processor;

a peripheral controller comprising first communications means that is capable of connecting the first processor to a set of peripheral resources and a second communications means that is capable of connecting the second processor to the same set of peripheral resources wherein each peripheral in the set of peripheral resources is capable of generating an interrupt signal;

the peripheral controller further comprising an interrupt controller that is capable of handling the interrupts from the set of peripheral resources and software interrupts and capable of independently generating a low priority interrupt signal and a high priority interrupt signal for each processor from the peripheral and software interrupts, the interrupt controller further comprising means for enabling the interrupt signals for the first processor, means for enabling the interrupt signals for the second processor, means for assigning a priority to the interrupt signals for the first processor and means for assigning a priority to the interrupt signals for the second processor wherein a particular interrupt signal is capable of being routed to either processor and is capable of being assigned a low priority or a high priority.